(1)

SOURCE: WESTINGHOUSE ELECTRIC CORPORATION
BALTIMORE, MD

CONTRACT NO. DAAK-70-82-C-0150

SPONORING ACTIVITY: ARMY NIGHT VISION AND ELECTRO-OPTICAL LABORATORY
FORT BELVOIR, VA

TITLE: VIDEO TRACKER MODELS - FINAL REPORT
DATE: FEB 83

VIDEO TRACKER MODELS

This preliminary report considers modelling of two classes of video trackers from their algorithmic and measurement aspects. Because they are widely used and easily implementable in hardware, the classes chosen were the binary centroid tracker and the binary correlation tracker. Many variations exist even in these classes; their value in the present setting is to demonstrate the modelling aspects and the salient features that may be derived from an analysis of the behavior of these systems in the presence of noise.

The models predict tracker offset errors (mean error) and track point "jitter" (the standard deviation of the error).

Track performance, in general, is dependent on the character of the object pattern being tracked and on the distribution of the noise associated in the formation of the array of pixels which represent the pattern. In the analyses which follow, a simple square pattern is used and Gaussian additive noise is assumed. The models permit any geometric pattern as input and the question of noise distribution is answered by the specification of the probabilities of detection and false alarm in the formation of the binary pattern from the linear data input.

Data is presented scaled by the pixel dimensions. Additional errors occur due to spatial quantization which, based on the assumption of uniformity, adds a variance of 1/12 pixel to each computed answer. This factor is included in the curves of rms error versus SNR; further, the curve data is the root sum square of the individual channel errors.

The salient features of the analyses are as follows:

- For the binary centroid tracker, the bias error is proportional to the track error.
- For the binary correlation tracker, the bias error is a function of the correlator search pattern.
- The correlation tracker is generally better than the centroid tracker. A curve of the RSS errors is shown in Figure 1.

BINARY CENTROID TRACKER

The analysis that follows provides an estimate of the error in the centroid measurement of a target pattern due to the presence of additive noise. Certain approximations and assumptions are evoked in order to ensure tractability of the mathematics.

The input to the tracker is assumed to be provided by a thresholded n by n matrix P of pixels derived from the imaging sensor output data. For a threshold level, T, the binary tracking pattern is defined as follows:

$$B_{ij} = \begin{cases} 1 & P_{ij} \geq T ; \quad i,j=1,\ldots,n \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

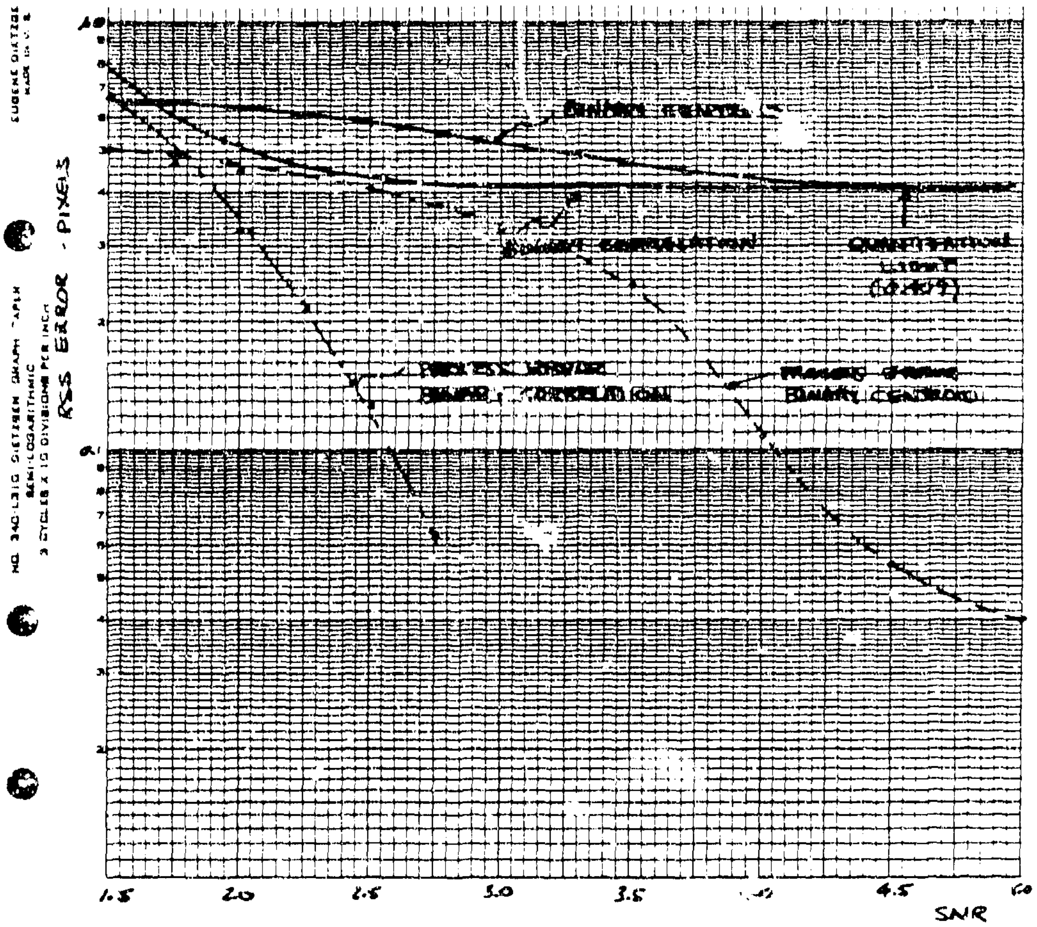

Figure 1.

The "one" values of the pattern result from two sources:

1) Target pattern detections with probability, P_t; and,

2) False alarm or background detections with probability, P_b.

Row and column vectors are formed from the binary matrix to compute the centroid.

Define

$$x_j = \sum_{i=1}^{n} B_{ij} \tag{2}$$

$$y_i = \sum_{j=1}^{n} B_{ij} \tag{3}$$

$$S = \sum_{i,j=1}^{n} B_{ij} \tag{4}$$

Given these forms, the centroid is defined as follows:

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{S} \tag{5}$$

$$\bar{y} = \frac{\sum_{i=1}^{n} w_i y_i}{S} \tag{6}$$

where

$$w_i = i - \left(\frac{n+1}{2}\right) \tag{7}$$

The values x_i, y_i, and S are random variables which are assumed to result from a series of independent Bernoulli trials. The number of "ones" that occur (or the integer value given) is described by a binomial probability law:

$$\text{Probability } [n=k] = \binom{m}{k} p^k (1-p)^{m-k} \tag{8}$$

which is the probability that exactly k ones occur of the possible m ones present. For this law, the mean is mp and the variance is mp(1-p). Now each element of the x vector is formed from target and background detections.

Let

$$t_j = \text{number of target elements in column } j \tag{9}$$

Then

$$b_j = n - t_j = \text{number of possible false alarm or background detections} \tag{10}$$

We will use the notation E(.) as the expectation operator. The variance Var(x) is defined as

$$\mathrm{Var}(x) = E(x^2) - E^2(x) \tag{11}$$

For the binomial law

$$E(x_j) = t_j P_t + (n-t_j) P_b \tag{12}$$

and

$$\mathrm{Var}(x_j) = t_j P_t (1-P_t) + (n-t_j) P_b (1-P_b) \tag{13}$$

Let

$$S_0 = \sum_{i=1}^{n} t_j = \text{target sum} \tag{14}$$

so that

$$E(S) = S_0 P_t + (n^2-S_0) P_b \tag{15}$$

$$\mathrm{Var}(S) = S_0 P_t (1-P_t) + (n^2-S_0) P_b (1-P_b) \tag{16}$$

(NOTE: Although there is partial correlation in forming the sum, independence is assumed.)

The centroid error, ϵ_x, is defined by

$$\epsilon_x = \bar{x} - \bar{x}_0 \tag{17}$$

where

$$\bar{x}_0 = \frac{\sum_{i=1}^{n} w_i t_i}{S_0} \tag{18}$$

and

$$E(\epsilon_x) = E(\bar{x}) - \bar{x}_0 \tag{19}$$

The computation of $E(\bar{x})$ involves the ratio of two random variables U and V, where

$$U = \sum_{i=1}^{n} w_i x_i \tag{20}$$

and

$$V = S \tag{21}$$

Assume that

$$U = U_0 + \eta$$

and

$$V = S_0 + \phi \tag{22}$$

where

$$E(\eta) = 0, \quad Var(\eta) = \sigma_U^2$$

$$E(\phi) = 0, \quad Var(\phi) = \sigma_V^2 \tag{23}$$

and

$$E(\eta.\phi) = E(\eta)E(\phi) = 0$$

Define

$$\bar{x} = \frac{U}{V} = \left(\frac{U_0}{V_0}\right) \frac{\left(1 + \frac{\eta}{U_0}\right)}{\left(1 + \frac{\phi}{V_0}\right)} \tag{24}$$

We invoke the approximation,

$$\bar{x} \doteq \left(\frac{U_0}{V_0}\right) \left(1 + \frac{\eta}{U_0} - \frac{\phi}{V_0}\right) \tag{25}$$

Then,

$$E(\overline{x}) = \frac{U_0}{V_0}\left(1 + \frac{E(\eta)}{U_0} - \frac{E(\phi)}{V_0}\right) \tag{26}$$

or

$$E(\overline{x}) = \frac{U_0}{V_0} \tag{27}$$

and

$$Var(\overline{x}) = \left(\frac{U_0}{V_0}\right)^2 E\left[\left(1 + \frac{\eta}{U_0} - \frac{\phi}{V_0}\right)^2\right] - E^2(\overline{x}) \tag{28}$$

$$\left(1 + \frac{\eta}{U_0} - \frac{\phi}{V_0}\right)^2 = 1 + \frac{2\eta}{U_0} - \frac{2\eta\phi}{U_0 V_0} - \frac{2\phi}{V_0} + \frac{\eta^2}{U_0^2} + \frac{\phi^2}{V_0^2} \tag{29}$$

From Equations (23)

$$Var(\overline{x}) = \left(\frac{U_0}{V_0}\right)^2 \left(\frac{\sigma_U^2}{U_0^2} + \frac{\sigma_V^2}{V_0^2}\right) \tag{30}$$

$$\sigma_V^2 = S_0 P_t (1-P_t) + (n^2-S_0) P_b (1-P_b) \tag{31}$$

$$V_0 = S_0 P_t + (n^2-S_0) P_b \tag{32}$$

$$U_0 = \sum_{i=1}^{n} w_i E(x_i) \tag{33}$$

$$U_0 = \sum_{i=1}^{n} w_i [t_i P_t + (n-t_i) P_b] \tag{34}$$

$$\sigma_U^2 = E\left[\left(\sum_{i=1}^{n} w_i x_i\right)^2\right] - U_0^2 \tag{35}$$

By independence,

$$\sigma_U^2 = P_t\,(1-P_t) \sum_{i=1}^{n} t_i w_i^2 + P_b(1-P_b) \sum_{i=1}^{n} (n-t_i)\, w_i^2 \tag{36}$$

$$E(\epsilon_x) = \frac{\sum_{i=1}^{n} w_i E(x_i)}{S_0 P_t + (n^2-S_0)\, P_b} - \frac{\sum_{i=1}^{n} w_i t_i}{S_0} \tag{37}$$

$$E(\epsilon_x) = \frac{\sum_{i=1}^{n} [S_0 w_i t_i P_t + S_0\,(n-t_i)\, w_i P_b - S_0 w_i t_i P_t - (n^2-S_0)\, P_b t_i]}{S_0 \quad S_0 P_t + (n^2-S_0)\, P_b} \tag{38}$$

$$E(\epsilon_x) = \frac{-\, n^2 P_b \sum_{i=1}^{n} w_i t_i}{S_0[S_0 P_t + (n^2-S_0)\, P_b]} \tag{39}$$

or

$$E(\epsilon_x) = \frac{-\, n^2 P_b \bar{x}_0}{S_0 P_t + (n^2-S_0)\, P_b} \tag{40}$$

This result indicates that when a centroid error results (i.e., $\bar{x}_0$ is not equal to zero), the process produces a biased answer which is proportional to the error and the probability of background detection.

From Equation (34),

$$U_0 = S_0\,(P_t-P_b)\,\bar{x}_0 \tag{41}$$

Now, to simplify examination of the variances, consider the 50 percent detection case with Gaussian additive noise.

Define

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-U^2/2}\, dU \tag{42}$$

Let the target have an amplitude A above the background and the background be Normal with mean μ and variance σ^2. The threshold is defined as

$$T = \mu + A/2 \tag{43}$$

From these definitions,

$$P_t = \frac{1}{\sqrt{2\pi}\sigma}\int_{\mu+N_2}^{\infty} e^{-\frac{(x-\mu-A)^2}{2\sigma^2}}\,dx \tag{44}$$

and

$$P_t = \frac{1}{\sqrt{2\pi}\sigma}\int_{\mu+N_2}^{\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx \tag{45}$$

from which,

$$P_t = 1 - \Phi\left(\frac{-A/2}{\sigma}\right) \tag{46}$$

$$P_t = 1 - \Phi\left(\frac{A/2}{\sigma}\right) \tag{47}$$

but since

$$\Phi(-x) = -\Phi(x) \tag{48}$$

$$P_t = \Phi\frac{A/2}{\sigma} = 1 - P_b \tag{49}$$

For this case,

$$P_t\,(1-P_t) = P_b\,(1-P_b) \tag{50}$$

so that from Equation (31),

$$\sigma_y^2 = n^2 P_t\,(1-P_t) \tag{51}$$

and

$$\sigma_U^2 = nP_t\,(1-P_t)\sum_{i=1}^{n} w_i^2 \tag{52}$$

$$w_i^2 = \left[i - \frac{n+1}{2}\right]^2 = i^2 - (n+1)\,i + \left(\frac{n+1}{2}\right)^2 \tag{53}$$

$$\sum_{i=1}^{n} w_i^2 = \frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)^2}{2} + n\left(\frac{n+1}{2}\right)^2 \tag{54}$$

$$= \frac{n(n+1)}{12}\,(4n+2 - 6n-6 + 3n+3) \tag{55}$$

$$\sum_{i=1}^{n} w_i^2 = \frac{n(n-1)^2}{12} \tag{56}$$

From Equation (30)

$$\mathrm{Var}(\bar{x}) = \frac{\sigma_U^2}{V_0^2} + \frac{U_0^2\sigma_V^2}{V_0^4} \tag{57}$$

and

$$\sigma_U^2 = \frac{n^2(n-1)^2}{12}\,P_t\,(1-P_t) \tag{58}$$

$$U_0 = S_0\bar{x}_0\,(2P_t-1) \tag{59}$$

$$V_0 = n^2\,(1-P_t) + S_0\,(2P_t-1) \tag{60}$$

$$\mathrm{Var}(\bar{x}) = n^2P_t(1-P_t)\left[\frac{(n-1)^2}{12\{n^2(1-P_t)+S_0(2P_t-1)\}^2} + \frac{S_0^2\bar{x}_0^2(2P_t-1)^2}{\{n^2(1-P_t)+S_0(2P_t-1)\}^4}\right] \tag{61}$$

Let the target to gate area be given by

$$R = \frac{S_0}{n^2} \tag{62}$$

and let n be large so that $(n-1)^2 \doteq n^2$

$$\mathrm{Var}(\overline{x}) = P_t(1-P_t)\left[\frac{1}{12\{1-P_t+R(2P_t-1)\}^2} + \frac{R^2\overline{x}_0^2(2P_t-1)^2}{\{1-P_t+R(2P_t-1)\}^4}\right] \tag{63}$$

For the tracking case, $\overline{x}_0$ is driven to zero so that the residual tracking noise approaches,

$$\mathrm{Var}(\overline{x}) \doteq \frac{P_t(1-P_t)}{12[1-P_t+R(2P_t-1)]^2} \tag{64}$$

Assuming that the detection probability ≥ 0.5, the form of the equation indicates that the variance decreases for either an increase in the detection probability or an increase in R. Since the detection probability is limited by signal-to-noise ratio, the result suggests that the gate be adjusted to accommodate the target and that the best result is obtained when R is maximized.

The modelling equations are summarized below:

$$\overline{x}_0 = \frac{\sum_{i=1}^{n} w_i t_i}{\sum_{i=1}^{n} t_i} \; ; \quad w_i = \left| i - \left(\frac{n+1}{2}\right)\right| \tag{65}$$

$$U_0 = (P_t - P_b)\, S_0\overline{x}_0 \tag{66}$$

$$V_0 = S_0P_t + (n^2-S_0)P_b \tag{67}$$

$$\sigma_V^2 = S_0P_t\,(1-P_t) + (n^2-S_0)P_b(1-P_b) \tag{68}$$

$$\sigma_U^2 = P_t\,(1-P_t)\sum_{i=1}^{n} t_i w_i^2 + P_b(1-P_b)\sum_{i=1}^{n}(n-t_i)\, w_i^2 \tag{69}$$

$$\varepsilon_x = \overline{x} - \overline{x}_0$$

$$E(\epsilon_x) = \frac{n^2 P_b \bar{x}_0}{S_0 P_t + (n^2 - S_0) P_b} \tag{70}$$

$$Var(\epsilon_x) = \frac{\sigma_U^2}{V_0^2} + \frac{U_0^2 \sigma_V^2}{V_0^4} \tag{71}$$

A Monte Carlo experiment was conducted using a 200-sample pattern of 8 by 8 squares in a 16 by 16 sample matrix. The signal-to-noise was established at the values shown in Table I and the statistics were compiled and tabulated (the pattern was shifted one pixel to the right and down).

Using the measured values of P_t and P_b and $x_0 \equiv 1$, the Equations (66) through (71) were used to compute the bias error $E(\epsilon_x)$ and the standard deviation of the centroid error,

$$\sqrt{Var(\epsilon_x)} \; .$$

These data are shown in Table II and indicate excellent agreement.

```
SNR=   1 [illegible]00
XBAR=   - [illegible]61   SIGMAX=     262
YBAR=     581   SIGMAY=     226
PT=     [illegible]   PB=    194
SNR=   [illegible] 000
XBAR=   - 424   SIGMAX=     33[illegible]
YBAR=     412   SIGMAY=     335
PT=     913   PB=    127
SNR=   [illegible] 500
XBAR=   - 276   SIGMAX=     293
YBAR=     310   SIGMAY=     298
PT=     887   PB=    083
SNR=   [illegible] 000
XBAR=   - 162   SIGMAX=     237
YBAR=   - 166   SIGMAY=     246
PT=     922   PB=    042
SNR=   [illegible] 500
XBAR=     080   SIGMAX=     179
YBAR=     077   SIGMAY=     165
PT=     951   PB=    021
SNR=   [illegible] 000
XBAR=   - 011   SIGMAX=     070
YBAR=   - 016   SIGMAY=     084
PT=     971   PB=    002
SNR=   4 500
XBAR=     003   SIGMAX=     035
YBAR=     004   SIGMAY=     031
PT=     987   PB=  0 000
SNR=   5 000
XBAR=   - 001   SIGMAX=     025
YBAR=     000   SIGMAY=     023
PT=     993   PB=  0 000
SNR=   [illegible] 500
XBAR=   - 000   SIGMAX=     014
YBAR=   - 002   SIGMAY=     015
PT=     997   PB=  0 000
SNR=   [illegible] 000
XBAR=     000   SIGMAX=     008
YBAR=     001   SIGMAY=     006
PT=   1 000   PB=  0 000
```

```
A TYPE B BINCEN FOR
        PROGRAM BINCEN
        DIMENSION IP(16,16) IQ(16,16) STATS(400 4)
        DIMENSION INOISE(2048)
        LOGICAL IPLINE(32),CLS
        COMMON PI
        PI=4 *ATAN(1 )
        NPTS=200
        PTS=FLOAT(NPTS)
        CLS=26
        WRITE(5,99) CLS
99      FORMAT(1X,A1)
        DO 5 I=1,32
5       IPLINE(I)=' '
        DO 10 I=1,16
        DO 10 J=1,16
        IP(I,J)=0
        IF((I GE 6) AND (I LE 13) AND (J GE 6) AND (J LE 13))
     1  IP(I,J)=64
10      CONTINUE
        DO 85 ISNR=1,10
        DO 86 I=1,NPTS
        STATS(I,3)=0
86      STATS(I,4)=0
        SNR=FLOAT(ISNR)/2 +1
        SIGMA=64 /SNR
        DC=3 *SIGMA
        SIGNAL=(SNR+3 )*SIGMA
        CALL GNOISE(DC,SIGMA,INOISE)
        IT=(DC+SIGNAL)/2
        DO 80 NPIX=1,NPTS
        DO 20 I=1,16
        DO 20 J=1,16
        IQ(I J)=0
        IF((IP(I,J)+IGAUSS(INOISE)) GT IT) IQ(I,J)=1
        IF(IQ(I,J) EQ 0) GOTO 20
        IF(IP(I,J) NE 0) STATS(NPIX,3)=STATS(NPIX,3)+1
        IF(IP(I,J) EQ 0) STATS(NPIX,4)=STATS(NPIX,4)+1
20      CONTINUE
        X=0
        Y=0
        SUM=0
        DO 30 I=1,16
        DO 30 J=1,16
        IF(IQ(I,J) EQ 0) GOTO 30
        X=X+FLOAT(J)
        Y=Y+FLOAT(I)
        SUM=SUM+1
30      CONTINUE
        XBAR=X/SUM-8 5
        YBAR=Y/SUM-8 5
        WRITE(5,99) CLS
        STATS(NPIX,1)=XBAR-1
        STATS(NPIX 2)=YBAR-1
        WRITE(5,200) XBAR,YBAR
200     FORMAT(2X 2F10 3)
        WRITE(5,250) NPIX
250     FORMAT(/2X 'BINARY IMAGE NUMBER' I3/)
        DO 40 I=1,16
```

```
      DO 45 J=1,16
      J1=2*J-1
      IPLINE(J1)=' '
      IF(IQ(I,J).EQ. ) IPLINE(J1)='X'
45    CONTINUE
40    WRITE(5,300) (   INE(J),J=1,32)
80    CONTINUE
      PT=0.
      PB=0.
      X1=0.
      Y1=0.
      X2=0.
      Y2=0.
      DO 60 I=1,NPTS
      PT=PT+STATS(I,3)
      PB=PB+STATS(I,4)
      X1=X1+STATS(I,1)
      X2=X2+STATS(I,1)**2
      Y1=Y1+STATS(I,2)
60    Y2=Y2+STATS(I,2)**2
      PT=PT/(64.*PTS)
      PB=PB/((256.-64.)*PTS)
      XBAR=X1/PTS
      YBAR=Y1/PTS
      SIGMAX=SQRT(X2/PTS-XBAR**2)
      SIGMAY=SQRT(Y2/PTS-YBAR**2)
      WRITE(5,400) XBAR,SIGMAX
      WRITE(5,410) YBAR,SIGMAY
      WRITE(2,420) SNR
      WRITE(2,400) XBAR,SIGMAX
      WRITE(2,410) YBAR,SIGMAY
      WRITE(2,430) PT,PB
85    CONTINUE
400   FORMAT(2X,'XBAR=',F8.3,2X,'SIGMAX=',F8.3)
410   FORMAT(2X,'YBAR=',F8.3,2X,'SIGMAY=',F8.3)
420   FORMAT(2X,'SNR=',F8.3)
430   FORMAT(2X,'PT=',F8.3,4X,'PB=',F8.3)
300   FORMAT(2X,32A1)
      END
C
      SUBROUTINE GNOISE(DC,SIGMA,INOISE)
      DIMENSION S(256),INOISE(2048)
      COMMON PI
      P1=1./SQRT(2.*PI)
      X=-.5
      H=.5/SIGMA
      DO 10 I=1,256
      A1=(X-DC)/SIGMA
      A2=(X+.5-DC)/SIGMA
      F1=EXP(-(A1*A1)/2.)
      F2=EXP(-(A2*A2)/2.)
      A3=(X+1.-DC)/SIGMA
      F3=EXP(-(A3*A3)/2.)
      S(I)=H*P1*(F1+4.*F2+F3)/3.
10    X=X+1
      M1=1
      DO 20 I=1,256
      N=2048.*S(I)
      M2=M1+N
      IF(M2.GT.2048) M2=2048
```

```
      DO 30 J=M1,M2
30    INOISE(J)=I-1
      M1=M2+1
20    CONTINUE
      RETURN
      END
C
      FUNCTION IGAUSS(INOISE)
      DIMENSION INOISE(2048)
      M=2048 *RAN(1.)+1.
      IGAUSS=INOISE(M)
      RETURN
      END
```

A > 50

1.500	-.567	.354
2.000	-.422	.328
2.500	-.292	.288
3.000	-.160	.231
3.500	-.083	.175
4.000	-.008	.077
4.500	0.000	.038
5.000	0.000	.028
5.500	0.000	.018

```
A>TYPE B:DISPR.FOR
      PROGRAM DISPR
      DIMENSION IPAT(16),VALS(2,2)
      DIMENSION V1(10),V2(10)
      COMMON PI
      DATA V1/.745,.823,.887,.922,.954,.971,.987,.993,.997/
      DATA V2/.184,.127,.083,.042,.021,.003,0.,0.,0./
      DATA IPAT/5*0,8*1,3*0/
      PI=4.*ATAN(1.)
      SNR=1.5
      DO 98 ISN=1,9
      S1=SNR/2.
      S2=-S1
      PT=V1(ISN)
      PB=V2(ISN)
      DO 99 NCASE=1,2
      H=8.
      IF(NCASE.EQ.2) H=0.
      AM=16.
      S=64.
      AMX=H*PT+(AM-H)*PB
      SIG2X=H*PT*(1.-PT)+(AM-H)*PB*(1.-PB)
      AMY=PT*S+(AM**2-S)*PB
      SIG2Y=S*PT*(1.-PT)+(AM**2-S)*PB*(1.-PB)
      S2X=SQRT(SIG2X)
      S2Y=SQRT(SIG2Y)
      R=AMX/AMY
      RSS=R*SQRT(SIG2X/(AMX*AMX)+SIG2Y/(AMY*AMY))
      VALS(1,NCASE)=R
      VALS(2,NCASE)=RSS*RSS
99    CONTINUE
      S1=0.
      S2=0.
      DO 20 I=1,16
      A=FLOAT(I)-8.5
      J=1
      IF(IPAT(I).EQ.0) J=2
      S1=S1+VALS(1,J)*A
20    S2=S2+VALS(2,J)*A*A
      S1=S1-1.
      S2=SQRT(S2)
      WRITE(2,100) SNR,S1,S2
100   FORMAT(1X,3F6.3)
98    SNR=SNR+.5
250   FORMAT(1H1)
      WRITE(2,250)
      END
C
      FUNCTION GAUSS(X)
      DOUBLE PRECISION B(5),P,T,T1,S
      COMMON PI
      DATA B/.31938153,-.356563782,1.781477937
     1 ,-1.821255978,1.330274429/
      P=.2316419
      Z=EXP(-(X*X)/2.)/SQRT(2.*PI)
      T=1./(1.+P*DBLE(ABS(X)))
      T1=T
      S=0
```

```
      DO 10 I=1,5
      S=S+T1*B(I)
10    T1=T1*T
      Q=SNGL(S)*Z
      IF(X.LT.0.) Q=1.-Q
      GAUSS=1.-Q
      RETURN
      END
C
      SUBROUTINE PRTVAL(LVAL)
      LOGICAL LVAL
10    IF((INP(X'EE').AND.1).NE.0) GOTO 10
      CALL OUT(X'EC',LVAL)
      CALL OUT(X'EF',14)
      CALL OUT(X'EF',15)
      RETURN
      END
```

```
A>TYPE B:DISPR.FOR
        PROGRAM DISPR
        DIMENSION IPAT(16),VALS(2,2)
        COMMON PI
        DATA IPAT/5*0,8*1,3*0/
        PI=4.*ATAN(1.)
        SNR=1.5
        DO 98 ISN=1,9
        S1=SNR/2.
        S2=-S1
        PT=1.-GAUSS(S2)
        PB=1.-GAUSS(S1)
        DO 99 NCASE=1,2
        H=8.
        IF(NCASE.EQ.2) H=0.
        AM=16
        S=64.
        AMX=H*PT+(AM-H)*PB
        SIG2X=H*PT*(1.-PT)+(AM-H)*PB*(1.-PB)
        AMY=PT*S+(AM**2-S)*PB
        SIG2Y=S*PT*(1.-PT)+(AM**2-S)*PB*(1.-PB)
        S2X=SQRT(SIG2X)
        S2Y=SQRT(SIG2Y)
        R=AMX/AMY
        RSS=R*SQRT(SIG2X/(AMX*AMX)+SIG2Y/(AMY*AMY))
        VALS(1,NCASE)=R
        VALS(2,NCASE)=RSS*RSS
99      CONTINUE
        S1=0.
        S2=0.
        DO 20 I=1,16
        A=FLOAT(I)-8.5
        J=1
        IF(IPAT(I).EQ.0) J=2
        S1=S1+VALS(1,J)*A
20      S2=S2+VALS(2,J)*A*A
        S1=S1-1.
        S2=SQRT(S2-S1*S1)
        WRITE(2,100) SNR,S1,S2
100     FORMAT(1X,3F6.3)
98      SNR=SNR+.5
250     FORMAT(1H1)
        WRITE(2,250)
        END
C
        FUNCTION GAUSS(X)
        DOUBLE PRECISION B(5),P,T,T1,S
        COMMON PI
        DATA B/.31938153,-.356563782,1.781477937
     1  ,-1.821255978,1.330274429/
        P=.2316419
        Z=EXP(-(X*X)/2.)/SQRT(2.*PI)
        T=1./(1.+P*DBLE(ABS(X)))
        T1=T
        S=0
        DO 10 I=1,5
        S=S+T1*B(I)
10      T1=T1*T
        Q=SNGL(S)*Z
```

```
      IF(X.LT.0.) Q=1.-Q
      GAUSS=1.-Q
      RETURN
      END
C
      SUBROUTINE PRTVAL(LVAL)
      LOGICAL LVAL
10    IF((INP(X'EE').AND.1).NE.0) GOTO 10
      CALL OUT(X'EC',LVAL)
      CALL OUT(X'EF',14)
      CALL OUT(X'EF',15)
      RETURN
      END
```

BINARY CORRELATION TRACKER

The correlation tracker to be analyzed below assumes the following form.

The tracker is a "searching type" wherein a reference is taken in one video frame and a picture sample is search in subsequent frames to determine a location where some sub-matrix in the picture sample best matches or maximally correlates to the reference. Errors are determined by noting where in the raster the picture sample is taken and where in the sample the best match occurs. The above description is served by an example which is used in subsequent experiments.

Suppose the picture sample is a 16 by 16 binary matrix and that at the initializing time, the reference is a centrally located 12 by 12 matrix within the picture sample. The matrices are shown in Figure 1. In subsequent frames

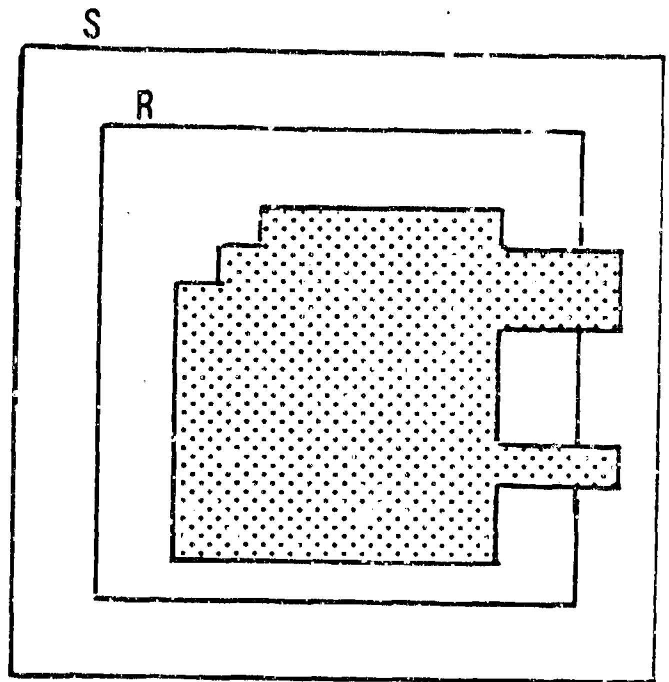

Figure 1.

the reference matrix is placed in the upper left hand corner of the picture sample, performs a correlation and is then moved one pixel to the right for the next correlation, etc. as shown in Figure 2.

In all, 25 positions of the reference will cover the picture sample. The resultant correlations are given in a 5 by 5 matrix C. The boresight position corresponds to C_{33}. Suppose maximum correlation occurs at C_{ij}. The errors are then given by

$$\begin{aligned} \epsilon_x &= j - 3 \\ \epsilon_y &= 3 - i \end{aligned} \qquad (1)$$

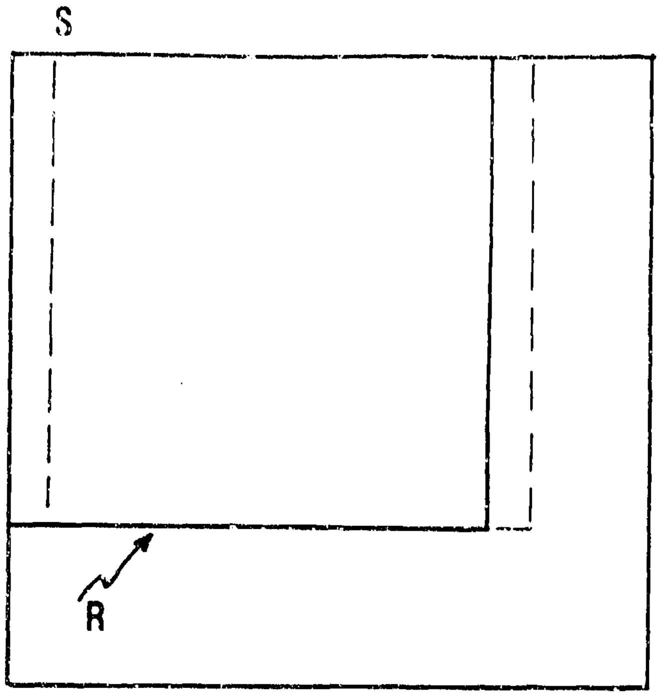

Figure 2.

(For this example, the dynamic range is $\pm$ 2 pixels although practical systems are designed to have a much larger dynamic range, typically $\pm$ 8 pixels.) The binary correlator produces true correlation measures with relatively simple processing as is shown below.

Suppose we are given two n vectors u and v. The correlation between u and v, $\rho(u,v)$ is defined as

$$\rho(u,v) = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2 \sum_{i=1}^{n} v_i^2}} \tag{2}$$

This calculation is rather formidable for real numbers; however, for the binary case we may assume that each number assumes only two states. For convenience, let

$$u_i, v_i \in \{-1,1\} \tag{3}$$

Then

$$u_i^2 = v_i^2 = 1 \tag{4}$$

and

$$\sqrt{\sum_{i=1}^{n} u_i^2 \sum_{i=1}^{n} v_i^2} = n \tag{5}$$

Now suppose that m of the elements of u match m of the elements of v one for one with $m \leq n$. Then,

$$\sum_{i=1}^{n} u_i v_i = m - (n-m) \tag{6}$$

That is, the remaining n-m elements have opposite sign. Consequently,

$$\rho(u,v) = \frac{2m - n}{n} \tag{7}$$

If all of the elements match, the correlation is one and if none of the elements match, the correlation is minus one. Now, working with conventional binary numbers; i.e.,

$$u_i, v_i \in \{0,1\}$$

$$\sum_{i=1}^{n} u_i v_i = n - 2 \sum_{i=1}^{n} i_i \oplus v_i \tag{8}$$

where the symbol $\oplus$ implies the "exclusive or" function. Then,

$$\rho(u,v) = \frac{2 * \text{number of "matches"}}{n} - 1 \tag{9}$$

The "exclusive or" truth table is as follows:

U	V	U $\oplus$ V
0	0	0
0	1	1
1	0	1
1	1	0

Now given the matrix R, we form a reference vector r as follows. Let

$$k = j + (i-1)\, n$$

and

$$r_k = C_{ij} \quad ; \quad i,j=1,\ldots,n \qquad (10)$$
$$; \quad k=1,2,\ldots,n^2$$

Similarly, define a sub-matrix of S as follows

$$T_{ij}^{k\ell} = S_{i+k-1,j+\ell-1} \qquad (11)$$

$$i,j=1,2,\ldots,16$$
$$k,\ell=1,2,\ldots,5 \qquad (12)$$

Then

$$t_m(k,\ell) = T_{ij}^{k\ell} = S_{i+k-1,j+\ell-1} \qquad (13)$$

where

$$m = j + 12(i-1) \qquad (14)$$

Then

$$C_{k\ell} = \rho\left(r, t(k,\ell)\right) \qquad (15)$$

The best correlation implies that

$$C_{k^*\ell^*} \geq C_{k\ell} \text{ for all } k,\ell=1,2,\ldots,5 \qquad (16)$$

and (k^*,ℓ^*) is the track point.

The basic correlation mechanism is rather straightforward; the analysis, however, is quite tedious. The search process produces a decision tree which must be evaluated to determine the estimated best point. As will be shown, the best point determination is biased according to the <u>order</u> in which the points are searched.

For the example given, 25 points must be searched. Let

$$P_k = \text{Prob}\left\{\text{position k is chosen}\right\} ; \quad k=1,\ldots,25 \qquad (17)$$

Further, let

$$R(m,n) = \text{Prob}\left\{\begin{array}{l}\text{correlation at position } m \\ \geq \text{correlation at position } n\end{array}\right\} \tag{18}$$

At the start of the process,

$$\begin{aligned} P_1 &= R(1,2) \\ P_2 &= R(2,1) \end{aligned} \tag{19}$$

and

$$R(2,1) = 1 - R(1,2) \tag{20}$$

with $n = 2$

The conditional probabilities can be computed by the following iteration:

$$P_{n+1} = \sum_{i=1}^{n} R(n+1,i)P_i \tag{21}$$

$$P_k = R(kmn+1)P_k \; ; \quad k=1,\ldots,n \tag{22}$$

$$n = n+1 \tag{23}$$

Equation (21) is interpreted as follows:

The $n+1$ state is chosen provided that the probability of the $n+1$ state is greater than some state i given that the state i had been previously chosen. Equation (22) implies that state k is chosen provided it indicates a greater probability than the current state and the probability that it had been previously chosen as the optimum state. The process stops when $n+1=m$ (or 25 in the current example). At any stage j in the process, the probability of choosing one of the states, P_T^j is given by

$$P_T^j = \sum_{i=1}^{j} P_i = 1 \tag{24}$$

$$P_T^j = \sum_{i=1}^{j} R(j-1,i)P_i + \sum_{i=1}^{j} R(i,j-1)P_i \; ; \quad j \geq 2 \tag{25}$$

$$P_T^j = \sum_{i=1}^{j} P_i[R(j-1,i) + R(i,j-1)] = \sum_{i=1}^{j} P_i = 1 \tag{26}$$

since

$$R(k,n) = 1 - R(n,k) \tag{27}$$

A partial diagram of the selection process is helpful. Consider four correlations:

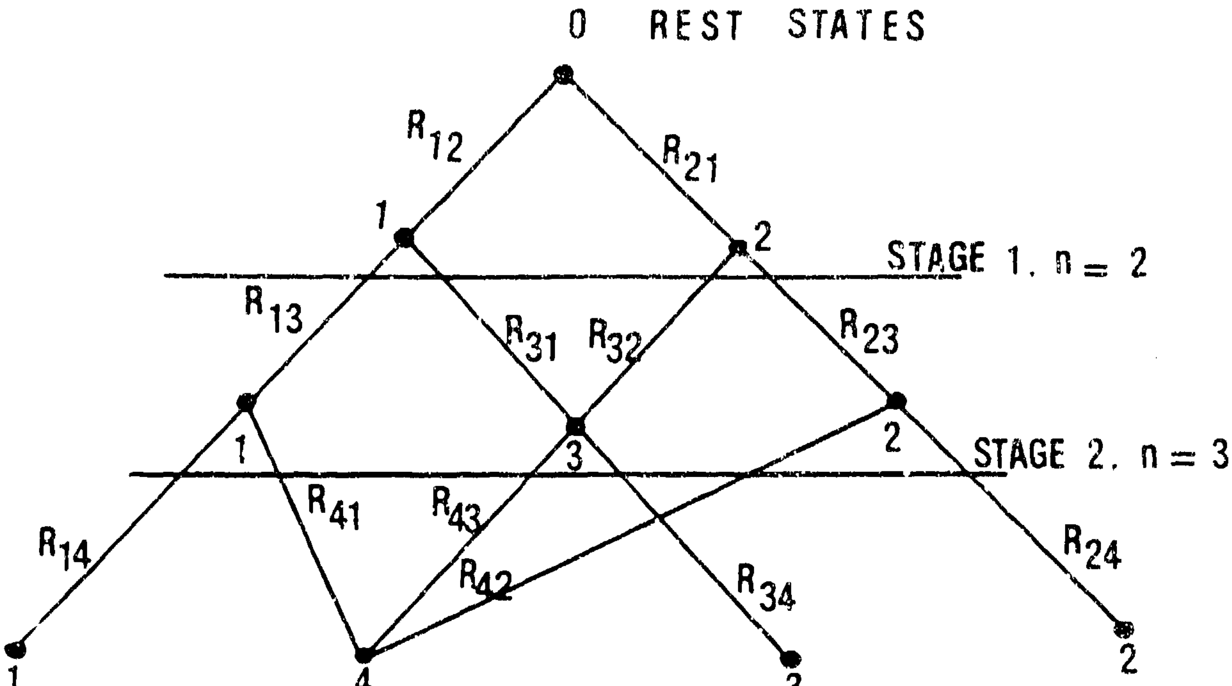

Figure 3.

$$\text{Prob}\left\{1 \text{ chosen}\right\} = P_1 = R_{14}R_{13}R_{12} \tag{28}$$

$$\text{Prob}\left\{2 \text{ chosen}\right\} = P_2 = R_{21}R_{23}R_{24} \tag{29}$$

$$\text{Prob}\left\{3 \text{ chosen}\right\} = P_3 = R_{34}(R_{31}R_{12}+R_{32}R_{21}) \tag{30}$$

$$\text{Prob}\left\{4 \text{ chosen}\right\} = P_4 = R_{41}R_{13}R_{12} + R_{43}(R_{31}R_{12}+R_{32}R_{21}) + R_{42}R_{23}R_{21} \tag{31}$$

$$P_T = P_1 + P_2 \quad P_3 + P_4 \tag{32}$$

$$\begin{aligned} P_T = {} & R_{12}(R_{14}R_{13}+R_{34}R_{31}+R_{41}R_{13}+R_{43}R_{31}) \\ & + R_{21}(R_{24}R_{23}+R_{34}R_{32}+R_{43}R_{32}+R_{42}R_{23}) \end{aligned} \tag{33}$$

$$P_T = R_{12} \; R_{13}(R_{14}+R_{41})+R_{31}(R_{34}+R_{43})$$
$$+ R_{21} \; R_{23}(R_{24}+R_{42})+R_{32}(R_{34}+R_{43}) \qquad (34)$$

$$P_T = R_{12}(R_{13}+R_{31}) + R_{21}(R_{23}+R_{32}) \qquad (35)$$

$$P_T = R_{12} + R_{21} = 1 \qquad (36)$$

Note that to search out this tree,

$$\frac{m(m-1)}{2} = 325$$

probabilities have to be determined (although some symmetries reduce this number).

The problem is to determine the probability that one position is better than another; however, this problem is complicated by the fact that the data in nearby positions is highly correlated. The algorithm used to compute the probabilities is shown in FORTRAN code in Figure 4, i.e., SUBROUTINE EVAL. The parameters k1, k2, k3, and k4 are reference position offsets. A diagram of the matrix positions is shown in Figure 5.

The algorithm computes the difference between the correlation values produced by the two reference positions (AM) and the variance of the difference measure (VAR). The probability that one position is better than another is the probability that the normalized difference value is greater than zero. The assumption is that

$$P(x) = \frac{1}{\sqrt{2\pi VAR}} \, e^{-\frac{(x-AM)^2}{2VAR}} \qquad (37)$$

$$PROB\,\{x > 0\} = \frac{1}{\sqrt{2\pi VAR}} \int_0^{\infty} e^{-\frac{(x-AM)^2}{2VAR}} \, dx \qquad (38)$$

$$PROB\,\{x > 0\} = \frac{1}{\sqrt{2\pi}} \int_{-\frac{AM}{\sqrt{VAR}}}^{\infty} e^{-\frac{u^2}{2}} \, du \qquad (39)$$

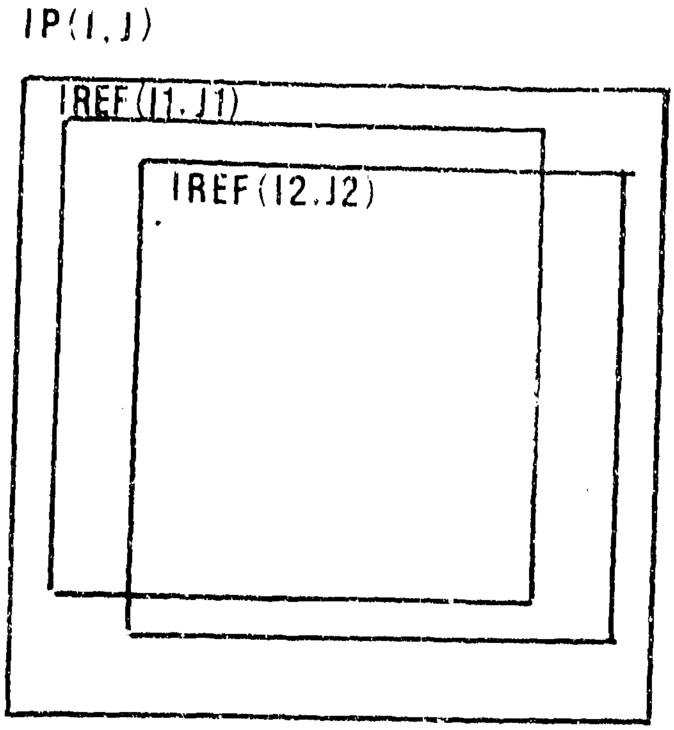

Figure 5.

and for

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{u^2}{2}} du \tag{40}$$

$$\text{PROB}\left\{x > 0\right\} \approx 1 - \Phi\left(\frac{-\text{AM}}{\sqrt{\text{VAR}}}\right) \tag{41}$$

The results of these computations are shown in Table III. The striking result is that the errors are biased according to the path taken by the search pattern; i.e., to the right and down.

To test the theory, a Monte Carlo test was run using 200 samples at various SNR. The results are shown in the following figures. The binary pattern shown corresponds to the initial reference. Data indicate that the errors are indeed biased. A 5 by 5 error matrix is also shown which indicates the number of times over the 200-sample run that a particular error position occurred (when divided by 200, this matrix approximates the probability matrices shown in Table III). Although the analysis is somewhat pessimistic, the agreement is good and the implied data trends are apparent. Another interesting result is that for rather moderate SNR ($\sim$ 2.5), the correlator essentially makes no error and can always correctly identify the correct target position; a significant error is made with the centroid measure for the same SNR.

```
A ...
A TYPE B BICOR FOR
      PROGRAM BICOR
      DIMENSION PROB2(5,5),IREF(12,12)
      DIMENSION PCHOSE(25),JLIST(25)
      DIMENSION PROB(25,25),IPOS(25,2)
      DIMENSION IP(16,16)
      COMMON PSQ
      DATA JLIST/0,0,0,0,0,2,0,0,0,0,0,8,9,0,0,0,4,7,
     1 14,0,0,5,10,11,20,0/
      PI=4.*ATAN(1.)
      PSQ=1./SQRT(2.*PI)
      DO 500 I=1,16
      DO 500 J=1,16
      IP(I,J)=0
      IF((I.GE.6).AND.(I.LE.13).AND.(J.GE.6).AND.(J.LE.13))
     1 IP(I,J)=1
500   CONTINUE
      DO 510 I=3,14
      DO 510 J=3,14
510   IREF(I-2,J-2)=IP(I,J)
550   FORMAT(//)
565   FORMAT(1H1)
      SNR=1.0
      DO 10 IPAR=1,6
      SIGMA=64./SNR
      DC=3.*SIGMA
      SIGNAL=(SNR+3.)*SIGMA
      IT=(DC+SIGNAL)/2.
      CALL GNOISE(DC,SIGMA,PT,PB,IT)
      WRITE(2,100) SNR,PT,PB
      DO 575 I=1,25
575   PROB(I,I)=0.
      DO 576 I=1,5
      DO 576 J=1,5
      N=J+5*(I-1)
      IPOS(N,1)=I
576   IPOS(N,2)=J
      DO 580 I=1,24
      I1=I+1
      DO 580 J=I1,25
      K1=IPOS(I,1)
      K2=IPOS(I,2)
      JJ=JLIST(J)
      IF((JJ.EQ.0).OR.(JJ.LT.I1)) GOTO 578
      P0=PROB(I,JJ)
      GOTO 579
578   L1=IPOS(J,1)
      L2=IPOS(J,2)
      CALL EVAL(IREF,IP,PB,PT,K1,K2,L1,L2,P0)
579   PROB(I,J)=P0
580   PROB(J,I)=1.-P0
      PCHOSE(1)=PROB(1,2)
      PCHOSE(2)=PROB(2,1)
      DO 700 I=3,25
      I1=I-1
      SUM=0
      DO 750 J=1,I1
      SUM=SUM+PROB(I,J)*PCHOSE(J)
      PCHOSE(J)=PROB(J,I)*PCHOSE(J)
```

```
  10      CONTINUE
          PCHOSE(I)=SUM
 700      CONTINUE
          DO 780 M=1,25
          I=1+(M-1)/5
          J=M-5*(I-1)
 780      PROBS(I,J)=PCHOSE(M)
          WRITE(2,789) ((PROBS(I,J),J=1,5),I=1,5)
 789      FORMAT(1X,5F9.5)
          AX=0
          AY=0
          VARX=0
          VARY=0
          DO 600 I=1,5
          DO 600 J=1,5
          ERRX=FLOAT(J-3)
          ERRY=FLOAT(I-3)
          AX=AX+PROBS(I,J)*ERRX
          VARX=VARX+PROBS(I,J)*(ERRX**2)
          AY=AY+PROBS(I,J)*ERRY
 600      VARY=VARY+PROBS(I,J)*(ERRY**2)
          VARX=SQRT(VARX-AX*AX)
          VARY=SQRT(VARY-AY*AY)
          WRITE(2,610) AX,VARX
          WRITE(2,620) AY,VARY
 610      FORMAT(4X,'XBAR=',F8.4,2X,'X STD  DEV =',F8.4/)
 620      FORMAT(4X,'YBAR=',F8.4,2X,'Y STD  DEV =',F8.4/)
  10      SNR=SNR+.5
 100      FORMAT(4X,'SNR=',F5.2,2X,'PT=',F7.3,2X,'PB=',F7.3)
          WRITE(2,565)
          END
C
          SUBROUTINE GNOISE(DC,SIGMA,PT,PB,IT)
          DIMENSION S(256)
          COMMON PSQ
          X=-.5
          H=.5/SIGMA
          DO 10 I=1,256
          A1=(X-DC)/SIGMA
          A2=(X+.5-DC)/SIGMA
          F1=EXP(-(A1*A1)/2.)
          F2=EXP(-(A2*A2)/2.)
          A3=(X+1.-DC)/SIGMA
          F3=EXP(-(A3*A3)/2.)
          S(I)=H*PSQ*(F1+4.*F2+F3)/3.
  10      X=X+1
          M1=1
          IS1=0
          IS2=0
          DO 20 I=1,256
          N=2048.*S(I)
          M2=M1+N
          IF(M2.GT.2048) M2=2048
          IF((I-1).GE.IT) IS1=IS1+M2-M1+1
          IF((I+63).GE.IT) IS2=IS2+1+M2-M1
          M1=M2+1
  20      CONTINUE
          PT=FLOAT(IS2)/2048
          PB=FLOAT(IS1)/2048
          RETURN
```

```
      END

      FUNCTION GAUSS(X)
      DOUBLE PRECISION B(5),P,T,S
      COMMON PSQ
      DATA B/.319381530,-.356563782,1.781477937,
     1-1.821255978,1.330274429/
      P=.2316419
      Z=EXP(-(X*X)/2.)*PSQ
      T=1./(1.+P*DBLE(ABS(X)))
      Q=SNGL(T*(B(1)+T*(B(2)+T*(B(3)+T*(B(4)+T*B(5))))))*Z
      IF(X.LT.0.) Q=1.-Q
      GAUSS=1.-Q
      RETURN
      END
C
      SUBROUTINE EVAL(IREF,IP,PB,PT,K1,K2,K3,K4,PR)
      DIMENSION IREF(12,12),IP(16,16)
      M1=1-K1
      M2=1-K2
      M3=1-K3
      M4=1-K4
      VARB=PB*(1.-PB)
      VART=PT*(1.-PT)
      AM=0
      VAR=0.
      DO 30 I=1,16
      DO 30 J=1,16
      I1=I+M1
      J1=J+M2
      I2=I+M3
      J2=J+M4
      L1=2
      L2=2
      IF((I1.GE.1).AND.(I1.LE.12).AND.(J1.GE.1).AND.(J1.LE.12))
     1 L1=IREF(I1,J1)
      IF((I2.GE.1).AND.(I2.LE.12).AND.(J2.GE.1).AND.(J2.LE.12))
     1 L2=IREF(I2,J2)
      IF((L2.EQ.2).AND.(L1.EQ.2)) GOTO 30
      M=IP(I,J)
      P=PT
      VARD=VART
      IF(M.EQ.0) P=PB
      IF(M.EQ.0) VARD=VARB
      IF(L1.NE.L2) GOTO 31
      VAR=VAR+2.*VARD
      GOTO 30
31    IF((L1.EQ.2).OR.(L2.EQ.2)) GOTO 35
      IF(L1.EQ.0) P=1.-P
      VARD=2.*VARD
      PR=2.*P-1
      GOTO 38
35    IF(L1.EQ.2) GOTO 37
      IF(L1.EQ.0) P=1.-P
      PR=P
      GOTO 38
37    IF(L2.EQ.0) P=1.-P
      PR=-P
38    AM=AM+PR
      VAR=VAR+VARD
```

```
 10   CONTINUE
      R=-AM/SQRT(VAR)
      PR=1.-GAUSS(R)
      WRITE(5,100) K1,K2,K3,K4,PR
100   FORMAT(1X,4I2,F8.4)
      RETURN
      END
```

```
SNR=  1.50   PT=    .754   PB=     .191
 .00000    .0000*    .00000    .0000*    .0000*
 .00000    .0000*    .00364    .00000    .0000*
 .00003    .00854    .72091    .03780    .0000
 .00001    .00101    .14819    .01460    .00019
 .00007    .00104    .04607    .01328    .00456
XBAR=     .0645   X STD. DEV.=     .3030
YBAR=     .2904   Y STD. DEV.=     .5860

SNR=  1.75   PT=    .800   PB=     .166
 .00000    .0000*    .0000*    .0000*    .0000*
 .00000    .0000*    .00323    .00000    .0000*
 .00001    .00654    .83015    .03004    .00002
 .00000    .00031    .10118    .00700    .00004
 .00001    .00018    .01720    .00337    .00072
XBAR=     .0349   X STD. DEV.=     .2222
YBAR=     .1483   Y STD. DEV.=     .4192

SNR=  2.00   PT=    .830   PB=     .132
 .00000    .0000*    .0000*    .0000*   0.00000
 .00000    .0000*    .00227    .00000    .0000*
 .00000    .00402    .90791    .01922    .00000
 .00000    .00007    .05822    .00256    .00000
 .00000    .00002    .00500    .00062    .00008
XBAR=     .0185   X STD. DEV.=     .1628
YBAR=     .0700   Y STD. DEV.=     .2848

SNR=  2.25   PT=    .863   PB=     .108
0.00000    .00000    .0000*    .0000*   0.00000
 .00000    .0000*    .00124    .00000   0.00000
 .00000    .00194    .95800    .00965    .00000
 .00000    .00001    .02736    .00067    .00000
 .00000    .00000    .00106    .00008    .00000
XBAR=     .0085   X STD. DEV.=     .1109
YBAR=     .0291   Y STD. DEV.=     .1816

SNR=  2.50   PT=    .894   PB=     .091
0.00000   0.00000    .00000   0.00000   0.00000
0.00000    .00000    .00053    .00000   0.00000
 .00000    .00075    .98395    .00387    .00000
 .00000    .0000*    .01060    .00013    .00000
 .00000    .0000*    .00016    .00001    .00000
XBAR=     .0033   X STD. DEV.=     .0689
YBAR=     .0105   Y STD. DEV.=     .1088

SNR=  2.75   PT=    .917   PB=     .071
0.00000   0.00000    .00000   0.00000   0.00000
0.00000   0.00000    .00015   0.00000   0.00000
 .00000    .00019    .99590    .00101    .00000
 .00000    .0000*    .00273    .00001    .00000
 .00000    .0000*    .00001    .00000    .0000*
XBAR=     .0008   X STD. DEV.=     .0348
YBAR=     .0026   Y STD. DEV.=     .0541
```

```
000000000000
000000000000
000000010000
000111111110
000111111110
000011111110
000011111110
000111111110
000111111100
001111111100
000011001100
000000000000


SNR=      1.500
XBAR=       .075   SIGMAX=        .591
YBAR=       .215   SIGMAY=        .528
PT=        .746    PB=         .184


    0     0     2     0     1
    1     0     0     0     0
    2    10   115    19     1
    2     0    35    10     1
    0     0     1     0     0
000000000000
000000000000
000000000000
000111111110
000111111110
000111111110
000111111110
000111111110
000111111110
000111111110
000001111100
000000000000


SNR=      2.000
XBAR=      -.010   SIGMAX=        .141
YBAR=       .025   SIGMAY=        .156
PT=        .824    PB=         .127


    0     0     0     0     0
    0     0     0     0     0
    1     0   194     0     0
    0     0     5     0     0
    0     0     0     0     0
000000000000
000000000000
000000000000
000011111110
000111111110
000111111110
000111111110
001111111110
000111111110
```

```
000111111110
000111111110
000000000000
```

```
SNR=      2.500
XBAR=      0.000   SIGMAX=     0.000
YBAR=      0.000   SIGMAY=     0.000
PT=       .887     PE=        .083
```

0	0	0	0	0
0	0	0	0	0
0	0	200	0	0
0	0	0	0	0
0	0	0	0	0